

MATERI AJAR 10 PEMROGRAMAN PHP

1.1. Materi Pembelajaran

Variabel, Fungsi dan Array di PHP

1.2. Kemampuan Akhir Pembelajaran

Sub CPMK 9 : Mahasiswa dapat memanfaatkan fungsi dan array untuk web dinamis

1.3. Materi Ajar

Variable

Apa itu variable? Variable adalah "containers" yang di gunakan untuk menampung sebuah value/nilai. Untuk membuat sebuah variable kamu hanya perlu menambahkan karakter (dolar) \$

Contohnya variable dalam PHP bisa kamu lihat di bawah ini:

```
<?php
$variable = "Hello Variable";
$variable_lain = "Hello Variable Lain";

echo $variable;
```

Tipe Data

Seperti bahasa pemrograman lainnya, PHP support beberapa tipe data yang bisa kamu gunakan dalam membuat sebuah variable. Adapun tipe data yang umum digunakan dalam PHP adalah:

Int, integer : semua angka (*numeric value*) tanpa koma ,

```
<?php
$x=1;
$y=2;
```

PHP

Double : angka dengan titik desimal atau angka dalam bentuk eksponensial.

```
<?php
$x = 1.22;
```

```
$y= 5.881230;
```

PHP

String : gabungan satu atau lebih karakter, seperti "hello world".

```
<?php
```

```
$string = "hello world!";
$string2 = "Hello world Lagi!";
```

PHP

Array : tipe data ini dapat digunakan untuk menampung beberapa nilai _(value) _sekaligus

```
<?php
```

```
$contoharray = array(1,2,4,'hello',);
```

PHP

_Booelan _** : **Tipe data ini hanya bernilai 2, yaitu *true* atau *false*, _kamu bisa menggunakannya untuk membuat sebuah conditional statement.

```
<?php
```

```
$x = true;
$y = false
$x = ($x==$y);
```

PHP

Object : Tipe data ini mirip dengan array, dapat digunakan untuk menampung beberapa nilai sekaligus tapi Object juga menampung informasi dan fungsi lain dari object tersebut. Untuk membuat sebuah object kamu perlu sebuah class dan mendeklarasikanya sebagai sebuah object.

```
<?php
```

```
class Mobil
{
    var $warna;
    function Mobil($warna="merah") {
        $this->warna = $warna;
    }
    function getWarna() {
```

```

return $this->warna;
}
}

$object_mobil = new Mobil;

```

PHP

`_NULL_`: tipe data ini hanya bernilai NULL atau kosong.

```
<?php
```

```
$x = NULL;
```

PHP

Operator

Selanjutnya kita akan membahas lebih jauh beberapa operator yang ada di dalam PHP dan penggunaannya.

Aritmatik Operator

Operator aritmatik berfungsi dalam fungsi matematika seperti tambah kuranga bagi dan yang lainnya.

Operator	Nama	Contoh Penggunaan
+	Addition (tambah)	$\$x + \y
-	Subtraction (kurang)	$\$x - \y
x	Multiplication (kali)	$\$x * \y
/	Division (bagi)	$\$x / \y
%	Modulus	$\$x \% \y
**	Exponentiation (pangkat)	$\$x ** 2$

Assignment Operator

Assignment operator berfungsi untuk memeberikan nilai ke sebuah variable.

- = : berfungsi untuk memberikan nilai ke atau dari variable ke variable lain. Contohnya $\$x = \$z + \$y$.
- += : berfungsi untuk penjumlahan/penambahan value. Contohnya $\$x += 5$, maksudnya adalah value dari $\$x$ akan ditambahkan dengan 5.
- -= : berfungsi untuk pengurangan. Contohnya $\$x -= 5$, maksudnya nilai dari $\$x$ dikurangi 5.
- *= : berfungsi untuk mengalikan. Contohnya $\$x *= 5$, maksudnya nilai dari $\$x$ akan dikalikan dengan 5.
- /= : berfungsi untuk membagi nilai. Contohnya $\$x /= 5$, maksudnya adalah nilai $\$x$ akan dibagi dengan 5.
- %= : berfungsi untuk mencari modulus dari sebuah variable. Contohnya $\$x \% = 5$, maka nilai $\$x$ akan dimodulus dengan 5.

Contoh penggunaan secara lengkap bisa kamu lihat di bawah ini:

```
<?php
$x=10;
echo $x; // outputs 10

$y=20;
$y += 100;
echo $y; // outputs 120

$z=50;
$z -= 25;
echo $z; // outputs 25

$i=5;
$i *= 6;
echo $i; // outputs 30

$j=10;
$j /= 5;
echo $j; // outputs 2

$k=15;
$k %= 4;
echo $k; // outputs 3
```

String Operator

- . (dot) : Concatenation atau rangkaian yang berfungsi untuk merangkai sebuah variable string dengan variable string lainnya.
- .= : Concatenation assignment yang juga berfungsi untuk merangkai string dengan string lainnya.

Contoh penggunaan String operator bisa kamu lihat di bawah ini:

```
<?php
$a = "Hello";
$b = $a . " world!";
echo $b; // outputs Hello world!

$x="Hello";
$x .= " world!";
echo $x; // outputs Hello world!
```

Increment / Decrement Operator

Operator ini berfungsi untuk menambahkan atau mengurangi sebanyak 1 dari jumlah variable sebelumnya.

- ++\$x : Pre-increment, jadi nilai \$x ditambahkan satu kemudian nilainya baru di return.
- \$x++ : Post-increment, jadi nilai \$x di return dulu kemudian baru nilainya ditambah 1.
- --\$x : Pre-decrement, jadi nilai \$x dikurangkan dulu dengan 1 baru direturn.
- \$x- : Post-decrement, jadi nilai \$x direturn dulu kemudian baru nilainya dikurangkan dengan 1.

Untuk bagaiman cara kerja operator di atas, kalian bisa cek lebih lanjut contoh kode dibawah ini:

```
<?php
$x=10;
echo ++$x; // outputs 11

$y=10;
echo $y++; // outputs 10

$z=5;
```

```
echo --$z; // outputs 4

$i=5;
echo $i--; // outputs 5
```

Comparison Operator

Operator ini digunakan untuk membandingkan 2 buah nilai. List operator yang tersedia bisa dilihat di bawah ini.

- == (Equal) : berfungsi membandingkan 2 buah value secara langsung, jika sama maka akan mereturn true, dan jika salah hasilnya adalah false.
- === (Identical) : berfungsi membandingkan 2 buah value dan juga sekaligus membandingkan apakah data tipenya sama atau tidak.
- != (Not Equal) : berfungsi untuk membandingkan apakah 2 buah nilai tersebut tidak sama.
- <> (Not Equal) : berfungsi sama dengan operator sebelumnya yaitu membandingkan apakah 2 nilai tersebut tidak sama.
- !== (Not identical)
- > (Lebih dari)
- < (Kurang dari)
- >= (Lebih besar dari atau sama dengan)
- <= (Lebih kecil dari atau sama dengan)

Untuk pemakaian comparison operator bisa di lihat di contoh di bawah ini:

```
<?php
$x=100;
$y="100";

var_dump($x == $y);
echo "<br>";
var_dump($x === $y);
echo "<br>";
var_dump($x != $y);
echo "<br>";
var_dump($x !== $y);
echo "<br>";
```

Logical Operator

Logical operator berfungsi untuk membandingkan dua buah value tapi yang dibandingkan adalah true dan false.

- `and` : akan menghasilkan true jika dua value yang di bandingkan sama-sama bernilai true
- `or` : akan bernilai true jika salah satu value dari yang di bandingkan bernilai true
- `xor` : akan bernilai true jikalau salah satu value dari yang di bandingkan bernilai true tapi tidak keduanya bernilai true.
- `&&` : fungsinya sama dengan operator ***and***
- `||` : fungsinya sama dengan operator ***or***
- `!` : berfungsi untuk mebalikkan value sebuah variable, bernilai true jika variable tersebut bernilai false

Untuk lebih jelasnya, kalian bisa lihat contoh kode di bawah ini:

```
<?php
$x = true;
$y = false;

var_dump( $x and $y ); // output false
var_dump( $x or $y ); // output true
var_dump( $x xor $y ); // output true
var_dump( $x && $y ); // output false
var_dump( $x || $y ); // output true
var_dump( !$x ); //output false
var_dump( !$y ); //output true
```

Array Operator

Array operator berfungsi untuk membandingkan 2 buah array secara langsung. Operatornya sama dengan comparison operator tapi variable yang di bandingkan memiliki tipe data array. Untuk lebih jelasnya kalian bisa lihat contoh kode di bawah ini:

```
<?php
$x = array("a" => "red", "b" => "green");
$y = array("c" => "blue", "d" => "yellow");
$z = $x + $y; // union of $x and $y
var_dump($z);
var_dump($x == $y);
var_dump($x === $y);
```

```
var_dump($x != $y);
var_dump($x <> $y);
var_dump($x !== $y);
?>
```

Function

Function adalah kumpulan dari beberapa statement PHP yang bisa di panggil dan digunakan berkali kali di dalam aplikasi kamu. Untuk memanggil sebuah function, kamu harus memanggilnya dengan namanya function tersebut secara langsung. contoh di bawah ini:

```
<?php
function myFirstFunction(){
    echo "Hello World!";
}

//cara memanggil fungsi di atas

myFirstFunction(); // akan menampilkan Hello World!
```

Contoh diatas adalah sebuah simple function dalam PHP. Jadi dalam membuat sebuah function kamu harus mengawalinya dengan statement function kemudian di ikuti dengan nama function tersebut. Sebenarnya dalam penamaan sebuah function, kalian bebas menentukan nama function.

Namun secara umum ada dua tipe yang biasa digunakan dalam penamaan di PHP.

Camel Case

Penamaan dengan camel case menggunakan huruf besar di setiap awal kata di dalam function tersebut. Biasanya namanya di mulai dengan huruf kecil lalu kemudian huruf besar di setiap awal kata atau setiap kata di dalam nama function itu dimulai dengan huruf besar. Contohnya *iniFungsiSaya*, *IniFungsiSaya*.

Snake Case

Berbeda dengan camel case, dengan style snake case, penamaan function di pisahkan oleh underscore (`_`) dalam setiap katanya. Contohnya adalah `ini_fungsi_saya` atau `INI_FUNGSI_SAYA`.

Function Parameters

Dalam membuat sebuah function, kalian dapat melewatkan sebuah value dari luar ke dalam function tersebut lewat sebuah parameter. Kalian bisa mendeklarasikan

seberapa banyak paramater yang kamu butuhkan sesuai dengan function yang kamu buat. Contoh function yang memiliki paramater adalah:

```
<?php
function functionWithParameter($param) {
    echo "Param value: ".$param;
}

functionWithParameter('Hello'); //output "Param value: Hello
```

Jike kita perhatikan contoh di atas maka kita bisa memarsing data dari luar ke dalam function tersebut dan jika kalian tidak memberikan sebuah parameter saat pemanggilan function tersebut maka code diatas akan menjadi error seperti contoh di bawah ini.

```
<?php
function functionWithParameter($param) {
    echo "Param value: ".$param;
}

functionWithParameter(); //dipanggil tanpa parameter

//error Warning: Missing argument 1 in call to
functionWithParameter()
```

Namun jika kamu ingin function tersebut dapat di panggil tanpa harus memerlukan sebuah parameter untuk di parsing, maka kamu perlu membuat parameter tersebut sebagai `_default argument value`. _Seperti apa maksudnya, coba perhatikan contoh di bawah ini.

```
<?php
function functionWithParameter($param = "Default Value") {
    echo "Param value: ".$param;
}

functionWithParameter(); //output -> Default Value

functionWithParameter('New Value'); //output -> New Value
```

Jadi parameter dalam function tersebut memiliki nilai parameter default yaitu `_Default Value_` dimana kalau saat pemanggilannya tanpa parameter maka nilai paramternya adalah yang menjadi default tersebut.

Jadi seperti itulah basic dalam pembuatan sebuah function dalam PHP, dengan membuat sebuah function untuk aplikasi yang kamu buat, maka kamu bisa mengurangi pengulangan code (*code duplication*) yang kamu tulis dan saat dibutuhkan kamu tinggal function tersebut lagi.

